PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 17/30 — A2

(11) International Publication Number: WO 00/51036

(43) International Publication Date: 31 August 2000 (31.08.00)

(21) International Application Number: PCT/US00/00868

(22) International Filing Date: 12 January 2000 (12.01.00)

(30) Priority Data:
09/259,620 26 February 1999 (26.02.99) US

(63) Related by Continuation (CON) or Continuation-in-Part (CIP) to Earlier Application
US 09/259,620 (CON)
Filed on 26 February 1999 (26.02.99)

(71) Applicant *(for all designated States except US)*: INTEL CORPORATION [US/US]; 2200 Mission College Boulevard, Santa Clara, CA 95052 (US).

(72) Inventors; and

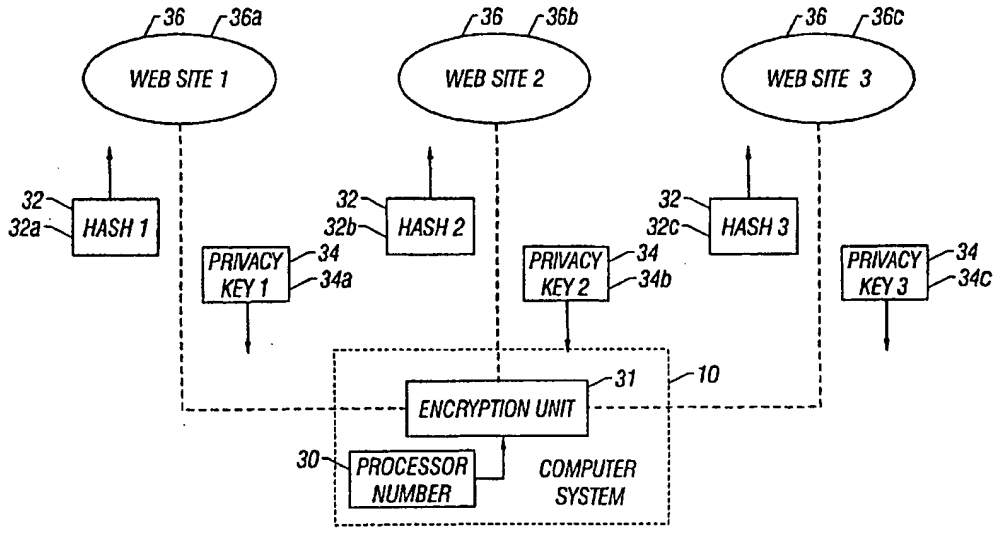
(75) Inventors/Applicants *(for US only)*: MI, James, Q. [CN/US]; 1361 Fisherhawk Drive, Sunnyvale, CA 94087 (US). PARIKH, Vishesh [IN/US]; 707 Continental Circle, #1034, Mountain View, CA 94040 (US). TENG, Albert, Y. [US/US]; 11597 Cedar Spring Court, Cupertino, CA 95014 (US).

(74) Agent: TROP, Timothy, N.; Trop, Pruner, Hu & Miles, P.C., Suite 100, 8554 Katy Freeway, Houston, TX 77024 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: COMPUTER SYSTEM IDENTIFICATION



(57) Abstract

A computer system (10) includes an interface (31) and a processor (200). The interface (31) is adapted to receive a request from another computer system for identification of the first computer system. The adapter (31) also furnishes a hash value that identifies the first computer system to the other computer system. The processor (200) is coupled to the interface (31) and is adapted to encrypt an identifier (30) that identifies the first computer system with a key (34) associated with the other computer system to provide the hash value (32).

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

Computer System Identification

Background

The invention relates to computer system identification.

A server (an Internet server, for example) may furnish a web site that provides a particular service. In this manner, a user of the web site may communicate with the web site via a client computer system. Sometimes the server may control access to the web site so that only a select group of users may access the service provided by the web site.

The ubiquitous use of e-mail and the rapid growth of community and chat-based web sites allow Internet users to reach out and interact with people whom they have never met. Unfortunately, not all individuals who participate in these forums are well-intentioned. Despite the fact that most chat-rooms use such access controls as a user name and password to control access to the community, a few users, after being denied access for inappropriate behavior, may circumvent these access controls. For example, a banned user may assume a new user name to regain access to the chat area and continue the disruptive behavior. This circumvention may allow single individuals to destroy the efforts of a group of people and lessen their enjoyment of the online experience.

An embedded identifier, such as a processor serial number (hereinafter called a "processor number"), may offer an effective means of deterring the above-described behavior by identifying the computer system that accesses the web site. For special chat rooms which require extra accountability, such as chat rooms for minors, web sites may create responsible chat environments where codes of conduct are enforceable and reliable by requiring that individuals provide their processor number (in addition to their name and password) to gain access to the chat room. If every member of a chat area volunteers his or her processor number, the net result is a more secure community that may more effectively deal with potentially threatening behavior. After all, if problem users volunteer to participate in a room that not only encourages but enforces responsible behavior via the use of processor numbers, their ability to regain denied access to the chat room may be thwarted, even if the problem users change their user names.

Unfortunately, the use of an embedded identifier to identify the client computer system may present difficulties. For example, the number may be used to build a trail of information that links the user to different databases that are maintained on the Internet.

These databases, in turn, may be correlated to build a vast database of private information about the user. Although, it is unlikely that such a database could be built given the fact that the user may have the capability to disable the processor number identification, there is a continuing need to enhance the user's privacy protection.

Summary

In one embodiment of the invention, a method includes receiving a request from a first computer system for identification of a second computer system and retrieving an identifier that identifies the second computer system. The identifier is encrypted with a key that is associated with the first computer system to produce a hash value. The hash value is provided to the first computer system in response to the request.

In another embodiment, a computer system includes an interface and a processor. The interface is adapted to receive a request from another computer system for identification of the first computer system. The interface furnishes a hash value that identifies the first computer system to the other computer system. The processor is coupled to the interface and is adapted to encrypt an identifier that identifies the first computer system with a key associated with the other computer system to produce the hash value.

In another embodiment, an article includes a storage medium readable by a first processor-based system. The storage medium stores instructions to cause a processor to receive a key from another system for identifying the first system and determine whether the key is valid. Based on the identification, the instructions cause the processor to selectively authorize encryption of an identifier that identifies the first system with the key to produce a hash value.

In yet another embodiment, a microprocessor includes an instruction unit, an execution unit and a bus interface unit. The instruction unit is adapted to indicate when the instruction unit receives an instruction that requests an identify that identifies the microprocessor. The execution unit is coupled to instruction unit and adapted to, in response to the indication from the instruction unit, encrypt a key with an identifier that identifies the microprocessor to produce a hash value. The bus interface unit is coupled to the execution unit and is adapted to furnish an indication of the hash value to external pins of the microprocessor.

Brief Description Of The Drawing

Fig. 1 is a schematic diagram of a network according to an embodiment of the invention.

Fig. 2 is an illustration of software executed by a computer system of Fig. 1 according to an embodiment of the invention.

Fig. 3 is a more detailed schematic diagram of a computer system of Fig. 1 according to an embodiment of the invention.

Fig. 4 is an illustration of the execution of an algorithm according to an embodiment of the invention to control identification requests by a processor of the computer system of Fig. 3.

Fig. 5 is a schematic diagram of the processor of Fig. 3 according to an embodiment of the invention.

Detailed Description

Referring to Fig. 1, an embodiment 10 of a computer system in accordance with the invention includes an encryption unit 31 that may receive identification requests from web sites 36 (web sites 36a, 36b and 36c, as examples) for an identity of the computer system 10. In response to these requests, the encryption unit 31 may furnish different hash values 32 (hash values 32a, 32b and 32c, as examples) to the different web sites 36. In some embodiments, each hash value 32 is different, and as a result, each web site 36 may identify the computer system 10 by a different hash value 32, although each of the hash values 32 is generated by a single processor number 30, as described below. Because each web site 36 associates the computer system 10 with a different hash value 32, information about a user of the computer system 10 may not be correlated between databases that are maintained by different web sites 36. For example, a particular web site 36 may identify the computer system 10 via the hash value "1bdf23" and another web site 36 may identify the computer system 10 via the hash value "53gh44." Furthermore, as described below, the manner in which the encryption unit 31 generates the hash values 32 makes it very difficult for a rogue web site 36 from obtaining the hash values 32 that identify the computer system 10 to other web sites 36. Therefore, due to the technique used by the encryption unit 31, it may be very different to correlate information about the user from databases that are maintained by different web sites 36. In this context, the term "web site" generally refers to an arrangement

where a computer system (a server, for example) executes software to provide a service to other computer systems, such as the computer system 10.

In the context of this application, the phrase "computer system" may generally refer to a processor-based system and may include (but is not limited to) a graphics system, a desktop computer, a mobile computer (a laptop computer, for example), or a set-top box as just a few examples. The term "processor" may refer to, as examples, at least one central processing unit (CPU), microcontroller, X86 microprocessor, Advanced RISC Machine (ARM) microprocessor or Pentium-based microprocessor. The examples listed above are not intended to be limiting, but rather, other types of computer systems and other types of processors may be included in some embodiments of the invention.

To obtain a hash value 32 that identifies the computer system 10, a particular web site 36 may transmit a privacy key 34 (privacy keys 34a, 34b and 34c, as examples) to the computer system 10. In response, the encryption unit 31 may encrypt an embedded identifier, such as a processor number 30, with the privacy key 34 to produce the hash value 32 that the computer system 10 furnishes to the requesting web site 36. In this manner, if each web site 36 transmits a different privacy key 34 to the computer system 10, then each web site 36 receives a different hash value 32, each of which indicates the computer system 10 to the particular web site 36. As described further below, the encryption unit 31 may include a processor 200 (see Fig. 3) to aid in the encryption of the privacy key 34 with the processor number 30.

The privacy key 34 may or may not be a private key, depending on the particular embodiment. For example, in some embodiments, the privacy key 34 may be derived from an address or universal resource locator (URL) for the web site 36. Therefore, as an example, the privacy key 34 may indicate a string, such as "www.example.com." As described below, for the embodiments where the privacy key 34 is derived from the URL, the computer system 10 may perform a validity check to determine if the privacy key 34 that is furnished by a particular web site 36 is based on the URL of the web site 36.

In some embodiments, the encryption unit 31 may use a hash function called F(PN, PRIVACYKEY) to perform the encryption. The F(PN, PRIVACYKEY) function may have properties that make it more difficult to track user information (about the computer system 10) that is stored on different web sites 36. For the F(PN, PRIVACYKEY) hash function, the

notation "PN" represents the processor number 30, and the notation "PRIVACYKEY" represents the privacy key 34.

One of the properties of the F(PN, PRIVACYKEY) hash function may be that the F(PN, PRIVACYKEY) function is a one way hash function, a notation that implies given the hash value 32 and the privacy key 34, it may be very difficult, if not impossible, to work backwards to determine the processor number 30. As a result, it may be very difficult for a particular web site 36 to use the hash value 32 that is obtained by that web site 36 to derive the processor number 30.

In some embodiments, another property of the F(PN, PRIVACYKEY) function may be that the F(PN, PRIVACYKEY) function is collision free, a term that means that it is highly unlikely for the F(PN, PRIVACYKEY) hash function to return the same hash value for different privacy keys 34. Thus, it may be highly unlikely for a particular website 36 to use the F(PN, PRIVACYKEY) function (with its associated privacy key 34) to obtain the same hash value 32 for two different processor numbers 30. Thus, this feature ensures that it is highly likely for a particular web site 36 to identify each computer system with a different, unique processor number 30.

Yet another property of the F(PN, PRIVACYKEY) function (in some embodiments) may be that the F(PN, PRIVACYKEY) function is non-commutative, as described below: $F(F(PN,PRIVACYKEY),PRIVACYKEY')) \neq F(F(PN,PRIVACYKEY'),PRIVACYKEY))$, where "PRIVACYKEY'" represents a privacy key 34 that is different from the privacy key 34 that is represented by "PRIVACYKEY." As a result of the non-commutative property, it may be very difficult to correlate the information that is associated with the computer system 10 (and user) on different databases (on different web sites 36) when different privacy keys 34 are used.

Many different hash functions may be used, in various embodiments, that satisfy one, more than one, or all of the properties described below. For example, in some embodiments, a secure hash algorithm (SHA), an algorithm that satisfies all of the properties described above, may be used.

In some embodiments, the computer system 10 may notify the user of the system 10 when a particular web site 36 is requesting system identification. For example, this notification may be in the form of a prompt in a window that is formed on a display 14 (see Fig. 3) of the computer system 10. In this manner, the user may either permit the web site 36

to obtain the identification (provided by the hash value 32) or reject the request. In some embodiments, the user may have an option to turn off the prompt.

Besides prompting the user about the identification request, the computer system 10 may take measures to prevent a rogue web site 36 from submitting an incorrect privacy key 34 for purposes of obtaining a hash value 32 that is associated with another web site 36. For example, in some embodiments, the request for identification may involve a two-part identification procedure. First, the web site 36 sets the privacy key 34 by executing (if authorized, as described below) an instruction (called SETKEY(PRIVACYKEY)) of the processor 200 (see Fig. 2). Referring to Fig. 2, as described below, the SETKEY(PRIVACYKEY) function may be associated with ring zero (i.e., the highest level) of an operating system 28. As a result, the computer system 10 may not permit execution of this processor instruction until the computer system 10 validates the provided privacy key 34 by executing a software program called a driver 19. After the privacy key 34 is validated by execution of the driver 19, the web site 36 may then be authorized to execute a processor instruction called HWID() (i.e., the HWID() instruction may not have an input parameter) that is associated with ring three (i.e., a lower privilege level) of the operating system 28 to obtain the hash value 32.

More particularly, in some embodiments, the above-described identification procedure may involve interaction between the operating system 28, an Internet browser 27 (Internet Explorer ® or Netscape Navigator ®, as examples) and the driver 19. For example, because the SETKEY(PRIVACYKEY) instruction is associated with ring zero, the web site 36 may not by itself cause execution of the instruction to obtain the hash value 32, as the web site 36 may only have access to ring three (a lower privilege level) and higher rings (i.e., even lower privilege levels) of the operating system 28. However, the driver 19 may have ring zero privileges and thus, may form a bridge between the web site 36 and the ring zero privileges of the operating system 28. In this manner, when the web site 36 attempts to execute SETKEY(PRIVACYKEY) instruction, the driver 19 may be called by the operating system 28 to cause the processor 200 to validate the privacy key 34 before providing the hash value 32. In the execution of the driver 19, the processor 200 may use results obtained from the execution of the browser 27 to validate the privacy key 34, as described below.

Referring to Fig. 4, when executed by the processor 200, the driver 19 may cause the processor 200 to perform the following functions. In particular, the driver 19 may cause the

processor 200 to determine (diamond 50) if the user has enabled an option to prompt the user when an identification request is received. If so, the processor 200 prompts (block 52) the user (via the display 14 (see Fig. 2), for example) that a web site 36 has submitted an identification request and waits for the user to indicate (via a keyboard 24 or move 26 (see Fig. 2), as examples) whether the user desires to reject the request. If so, the processor 200 rejects the request by notifying (block 56) the web site 36.

However, if the user did not reject the request, then the processor 200 may determine (diamond 58) whether the browser 27 is currently being executed. If so, the program 19 causes the processor 200 to communicate (block 60) the privacy key 34 to the browser 27 so that when the processor 200 executes the browser 27 (on another thread, for example), the processor 200 may compare the URL of the web site 32 to the privacy key 34. Subsequently, the processor 200, communicates the results of the comparison for use by the driver 19. In this manner, when the processor 200 subsequently executes the driver 19, the processor 200 determines (diamond 62) whether the privacy key 34 matches the URL of the web site 36. If not, the processor 200 rejects the request and notifies (block 56) the web site 36 about the rejection of the identification request. Otherwise, the processor 200 executes (block 64) the SETKEY(PRIVACYKEY) instruction to set the privacy key to be used for the encryption of the processor number 30. In this manner, the web site 36 that submitted the privacy key 34 may cause the processor 200 to execute the HWID() instruction to cause the processor 200 to produce an indication of the hash value 32. However, if the privacy key 34 has not been set, then the processor 200 returns an indication of an error rather than the indication of the hash value 32.

Referring back to Fig. 3, in some embodiments, the computer system 10 may include a bridge, or memory hub 16. The processor 200 and the memory hub 16 may be coupled to a host bus 23. The memory hub 16 may provide interfaces to couple the host bus 23, a memory bus 29 and an Accelerated Graphics Port (AGP) bus 11 together. The AGP is described in detail in the Accelerated Graphics Port Interface Specification, Revision 1.0, published on July 31, 1996, by Intel Corporation of Santa Clara, California. The system memory 18 may be coupled to the memory bus 29, and store the driver 19, the browser 27 and portions of the operating system 28 (not shown in Fig. 3). A graphics accelerator 13 (that controls the display 14) may be coupled to the AGP bus 11. A hub communication link 15 may couple the memory hub 16 to another bridge circuit, or input/output (I/O) hub 20.

In some embodiments, the I/O hub 20 includes interfaces to an I/O expansion bus 25 and a Peripheral Component Interconnect (PCI) bus 21. The PCI Specification is available from The PCI Special Interest Group, Portland, Oregon 97214. A network interface 12 (a modem or a local area network (LAN) card, as examples) may be coupled to the PCI bus 21 and provide a communication path for the computer system 10 to communicate with the web sites 36. In this manner, the processor 200 may interact with the network interface 12 to communicate with the web sites 32. The I/O hub 20 may also include interfaces to a hard disk drive 37 and a CD-ROM drive 33, as examples. An I/O controller 17 may be coupled to the I/O expansion bus 25 and receive input data from the keyboard 24 and the mouse 26, as examples. The I/O controller 17 may also control operations of a floppy disk drive 22. Copies of the driver 19 may be stored on, as examples, the hard disk drive 32, a diskette or a CD-ROM, as just a few examples.

Referring to Fig. 5, as an example, the processor 200 may include a bus interface unit (BIU) 208 that is coupled to address, control and data lines of the host bus 23 to, among other operations, retrieve instructions and data from the system memory 18. For the instructions, the processor 19 may include an instruction unit 203 that is coupled to the bus unit 208 to decode the instructions. In this manner, the instruction unit 203 may have buffers and a cache to store the instructions. A control unit 208 (of the processor 200) may receive signals from the instruction unit 203 that indicate the decoded instructions. For example, the signals may indicate the instruction to perform the SETKEY(PRIVACYKEY) function or the instruction to perform the HWID() function.

In response to the instruction that is indicated by the instruction unit 203, in some embodiments, the control unit 208 may retrieve corresponding elementary instructions, called microcode, from a microcode read only memory (ROM) 210 of the processor 200 and execute the microcode. For example, microcode 211 to cause the processor 200 to perform the SETKEY(PRIVACYKEY) and HWID() instructions may be stored in a microcode read only memory (ROM) 210. In performing the execution of an instruction, the control unit 208 may control an arithmetic logic unit (ALU) 212, registers 214 and an addressing unit 206.

Other embodiments are within the scope of the following claims. For example, in other embodiments, the circuitry to perform the SETKEY(PRIVACYKEY) and HWID() instructions may be hardwired instead of being implemented in microcode. The processor number 30 may be replaced by another identifier that identifies the computer system 10. A

privacy key other than a string that indicates an URL may be used. Applications other than applications being executed by web sites may request identification of the computer system 10. For example, other computer systems that are connected through a local area network (LAN) may request identification from the computer system 10.

While the invention has been disclosed with respect to a limited number of embodiments, those skilled in the art, having the benefit of this disclosure, will appreciate numerous modifications and variations therefrom. It is intended that the appended claims cover all such modifications and variations as fall within the true spirit and scope of the invention.

What is claimed is:

1. A method comprising:

receiving a request from a first computer system for identification of a second computer system;

retrieving an identifier that identifies the second computer system;

encrypting the identifier with a key associated with the first computer system to produce a hash value; and

providing the hash value to the first computer system in response to the request.

2. The method of claim 1, wherein the act of retrieving the identifier comprises:

retrieving a processor number that identifies a processor of the second computer system.

3. The method of claim 2, further comprising:

executing a processor instruction; and

retrieving the number in response to the execution of the instruction.

4. The method of claim 1, further comprising:

receiving the key from the first computer system.

5. The method of claim 1, wherein the key indicates an address of a web site.

6. A computer system comprising:

an interface adapted to:

receive a request from another computer system for identification of the first computer system, and

furnish a hash value that identifies the first computer system to said another computer system; and

a processor coupled to the interface and adapted to:

encrypt an identifier that identifies the first computer system with a key associated with said another computer system to produce the hash value.

7. The computer system of claim 6, wherein the identifier comprises a processor number that identifies the processor.

8. The computer system of claim 6, wherein the processor comprises:
a memory adapted to store microcode for performing the encryption; and
a control unit coupled to the memory and adapted to execute the microcode to perform the encryption.

9. The computer system of claim 6, wherein the processor is further adapted to:
interact with the interface to receive the key from said another computer system.

10. An article comprising a storage medium readable by a first processor-based system, the storage medium storing instructions to cause a processor to:
receive a key from another processor-based system for identifying the first system,
determine whether the key is valid,
based on the identification, selectively authorize encryption of an identifier that identifies the first system with the key to produce a hash value.

11. The article of claim 10, the storage medium storing instructions to cause the processor to:
use an address of said another system to determine whether the key is valid.

12. The article of claim 11, wherein the key indicates an URL address.

13. The article of claim 10, the storage medium storing instructions to cause the processor to:
execute an instruction to cause the processor to subsequently use the key to produce the hash value.

14. The article of claim 10, wherein the identifier comprises a processor number.

15. A microprocessor comprising:

an instruction unit adapted to indicate when the instruction unit receives an instruction that requests an identifier that identifies the microprocessor;

an execution unit coupled to the instruction unit and adapted to, in response to the indication from the instruction unit, encrypt a key with the identifier to produce a hash value; and

a bus interface unit coupled to the execution unit and adapted to furnish an indication of the hash value to external pins of the microprocessor.

16. The microprocessor of claim 15, wherein the execution unit comprises:

a control unit coupled to the algorithmic unit and the registers; and

a memory coupled to the control unit and storing microcode to cause the control unit to use the key and the identifier to produce the hash value.

17. The microprocessor of claim 15, wherein the identifier comprises a processor number.

18. The microprocessor of claim 15, wherein the execution unit is adapted to use a one way hash function to produce the hash value.

19. The microprocessor claim 15, wherein the execution unit is adapted to use a non-commutative hash function to produce the hash value.

20. The microprocessor of claim 15, wherein the execution unit is adapted to use a collision free hash function to produce the hash value.

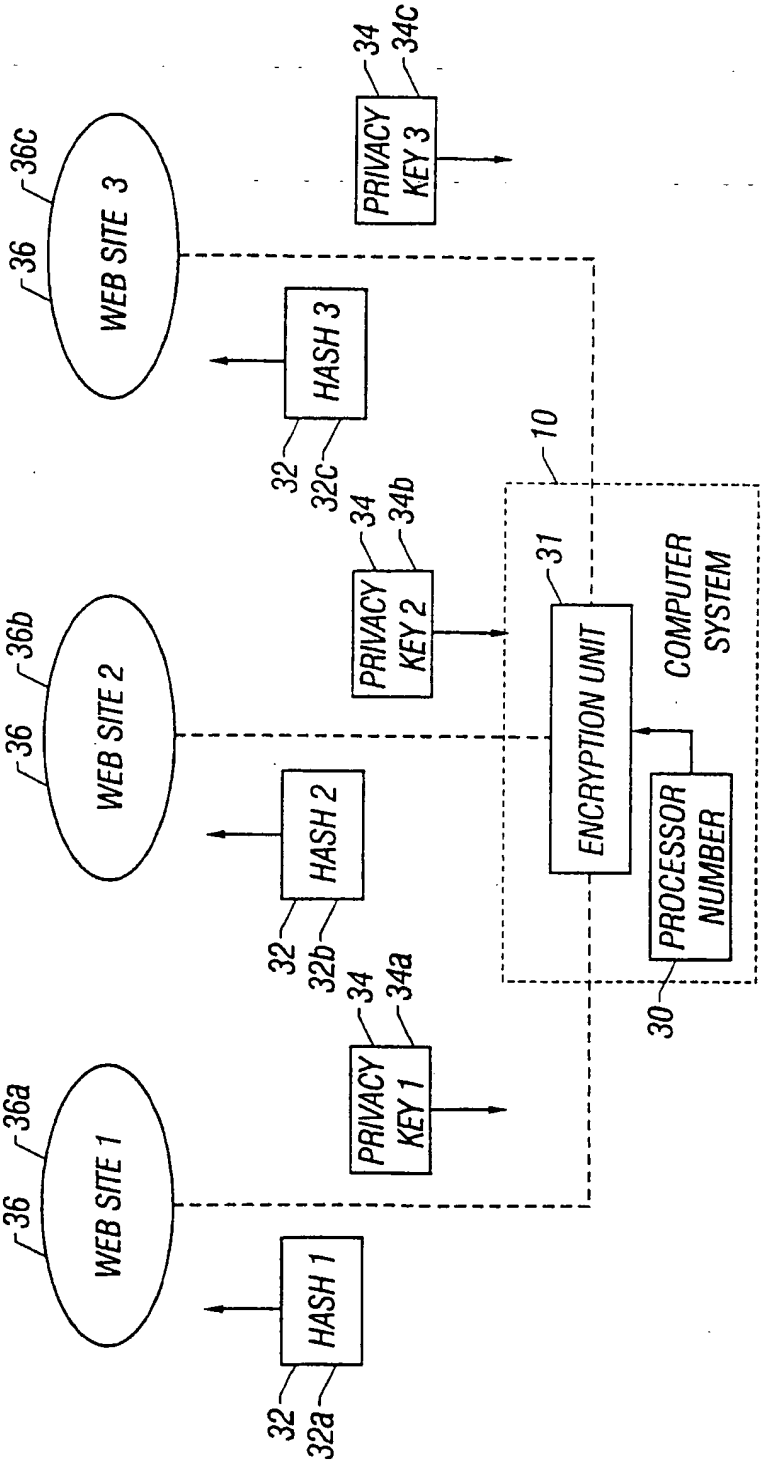

WEB SITE 1
WEB SITE 2
WEB SITE 3
36
36a
36b
36c
HASH 1
HASH 2
HASH 3
32
32a
32b
32c
PRIVACY KEY 1
PRIVACY KEY 2
PRIVACY KEY 3
34
34a
34b
34c
ENCRYPTION UNIT
31
PROCESSOR NUMBER
COMPUTER SYSTEM
10
30

FIG. 1

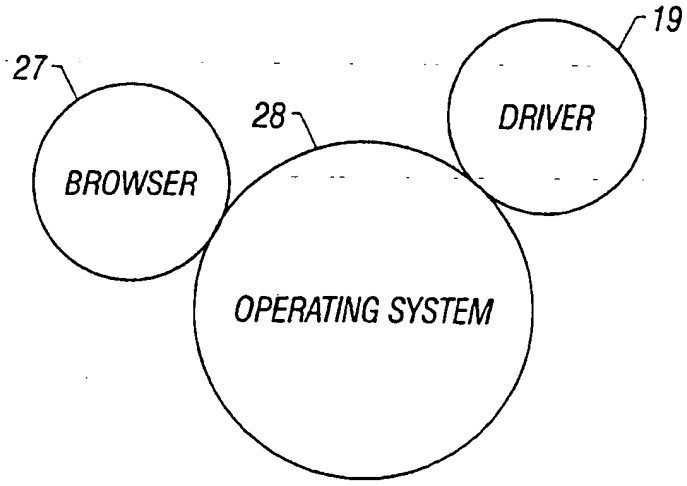

19
27
28
DRIVER
BROWSER
OPERATING SYSTEM

FIG. 2

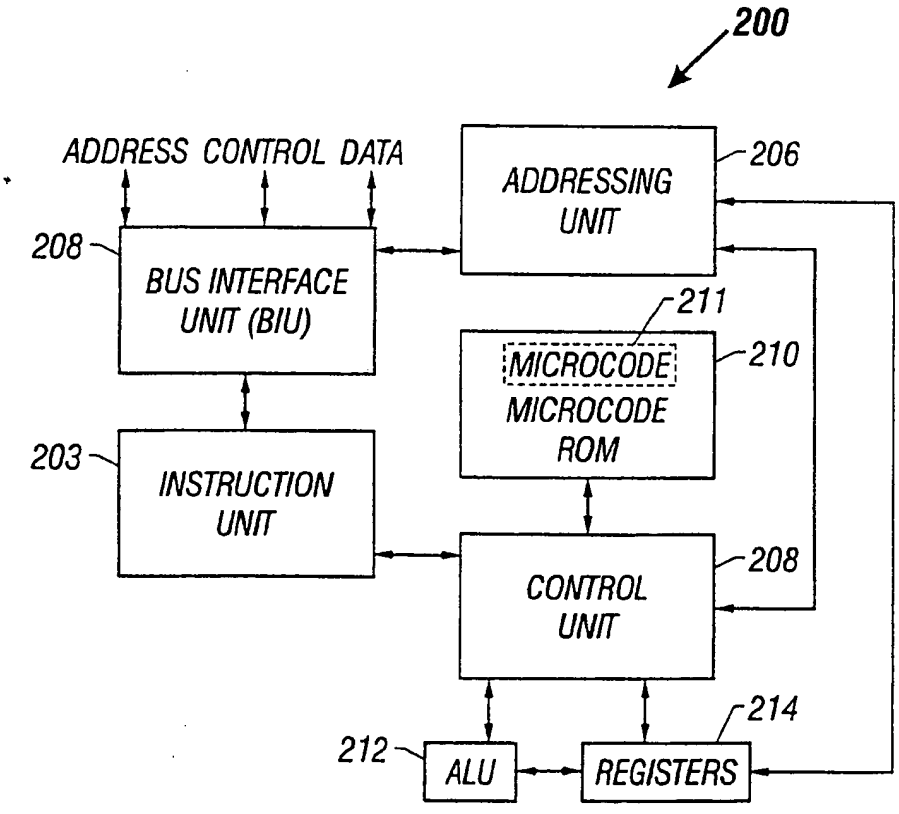

200
ADDRESS CONTROL DATA
206
ADDRESSING UNIT
208
BUS INTERFACE UNIT (BIU)
211
MICROCODE
210
MICROCODE ROM
203
INSTRUCTION UNIT
208
CONTROL UNIT
214
212
ALU
REGISTERS

FIG. 5

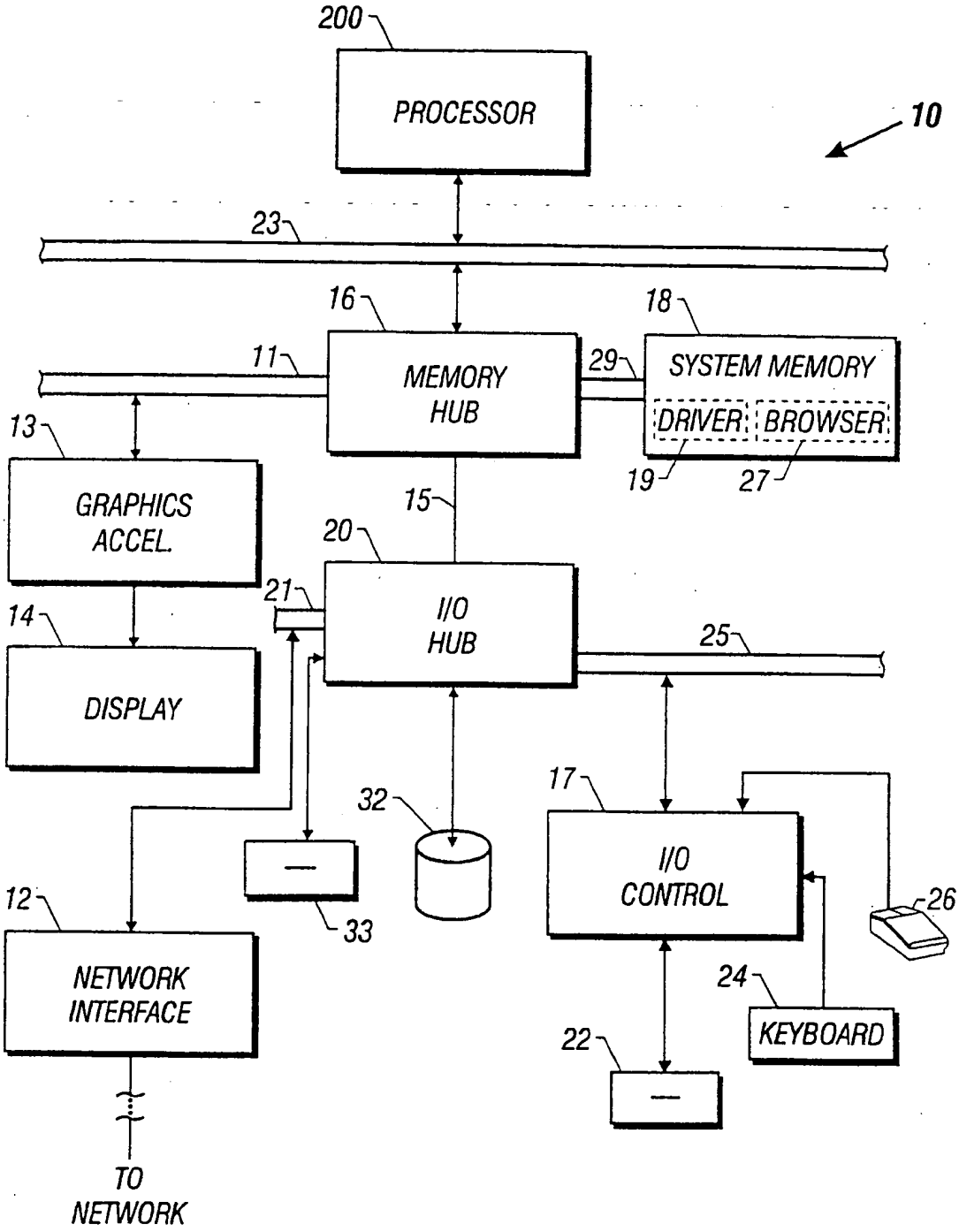

200
PROCESSOR
10
23
16
18
11
MEMORY HUB
29
SYSTEM MEMORY
DRIVER
BROWSER
13
19
27
GRAPHICS ACCEL.
15
20
21
I/O HUB
25
14
DISPLAY
32
17
I/O CONTROL
26
12
33
NETWORK INTERFACE
24
KEYBOARD
22
TO NETWORK

FIG. 3

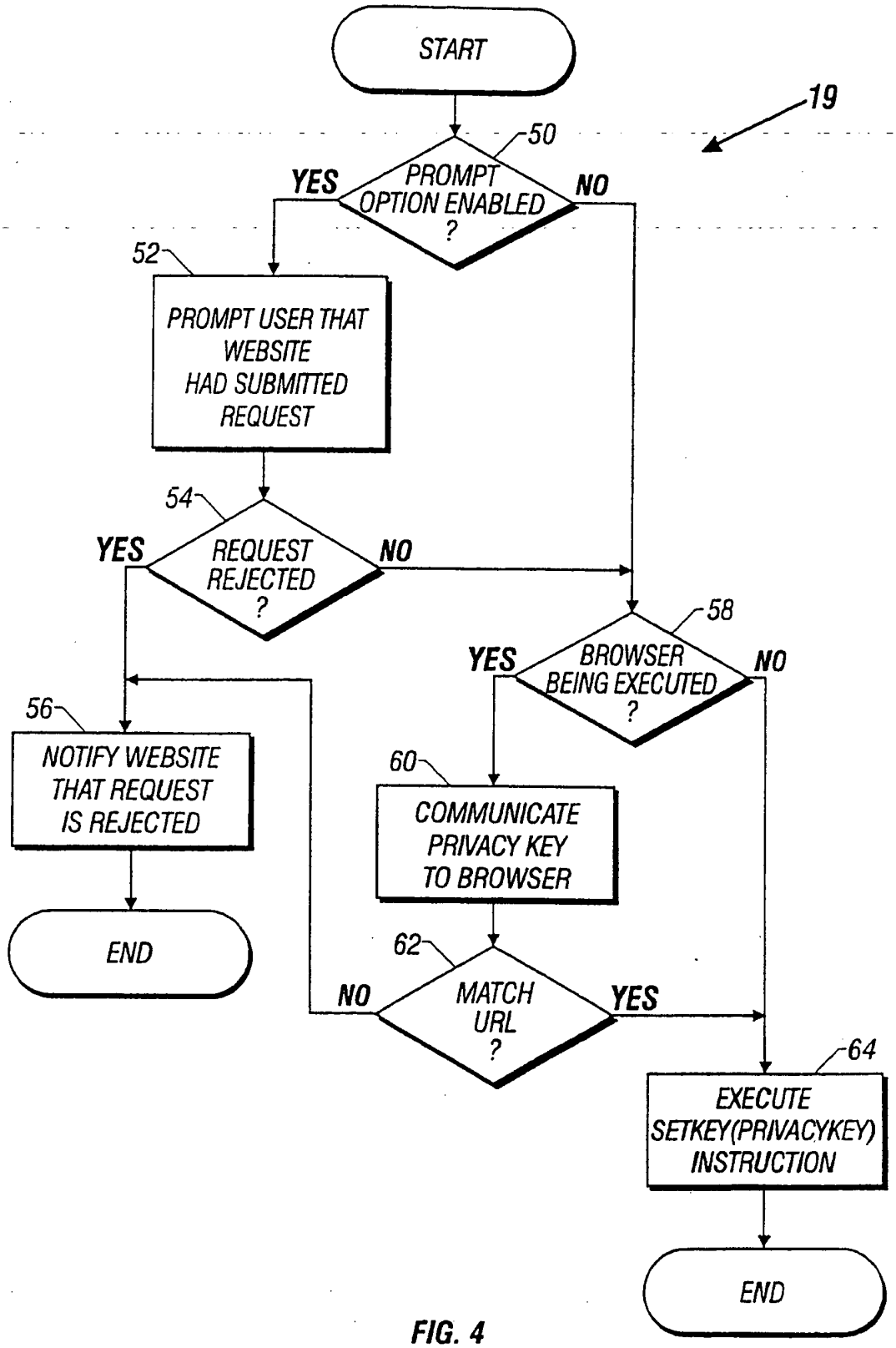

START
19
50
PROMPT OPTION ENABLED ?
YES
NO
52
PROMPT USER THAT WEBSITE HAD SUBMITTED REQUEST
54
REQUEST REJECTED ?
YES
NO
58
BROWSER BEING EXECUTED ?
YES
NO
56
NOTIFY WEBSITE THAT REQUEST IS REJECTED
END
60
COMMUNICATE PRIVACY KEY TO BROWSER
62
MATCH URL ?
NO
YES
64
EXECUTE SETKEY(PRIVACYKEY) INSTRUCTION
END

FIG. 4